



UNISUL

UNIVERSIDADE DO SUL DE SANTA CATARINA

GUILHERME BREGOLIN

**PROPOSTA DE APLICATIVO PARA ACADEMIAS PARA DISPOSITIVOS
ANDROID**

Curitiba, Campus Unisul Virtual

2020

GUILHERME BREGOLIN

**PROPOSTA DE APLICATIVO PARA ACADEMIAS PARA DISPOSITIVOS
ANDROID**

Relatório de pesquisa na modalidade de Estudo de Caso apresentado ao Curso de **Tecnólogo em Gestão da Tecnologia da Informação** da Universidade do Sul de Santa Catarina, como requisito parcial à aprovação na unidade de aprendizagem de Estudo de Caso.

Orientador: Prof. Me. Nilce Miranda Ayres.

Curitiba, Campus Unisul Virtual

2020

GUILHERME BREGOLIN

**PROPOSTA DE APLICATIVO PARA ACADEMIAS PARA DISPOSITIVOS
ANDROID**

Este trabalho de pesquisa na modalidade de Estudo de Caso foi julgado adequado, em sua forma final, à aprovação na unidade de aprendizagem de Estudo de Caso, do curso de **Tecnólogo em Gestão da Tecnologia da Informação** da Universidade do Sul de Santa Catarina.

Curitiba, 16 de novembro de 2020

Professor e orientador, Me. Nilce Miranda Ayres.
Universidade do Sul de Santa Catarina

AGRADECIMENTOS

Agradeço a minha esposa pelo apoio incondicional. Agradeço ao amigo Murilo pelo expertise acadêmico e paciência.

RESUMO

O presente trabalho teve como tema proposta de um aplicativo para academias para dispositivos Android. O método adotado na formulação deste trabalho, encontra-se em concordância com a proposta de estudo, a qual encontra-se adequada por meio dos objetivos a serem alcançados. O desenvolvimento da ciência tem como base o alcance de resultados que permite validar hipóteses sobre determinado acontecimento ou fato, presente em nossas vidas, ou não. Assim, o objetivo geral desta pesquisa, busca apresentar uma proposta voltada para o desenvolvimento de um aplicativo Android para academias. Os objetivos específicos do presente trabalho busca apresentar e conceituar o que é um dispositivo móvel, bem como salientar o que é a plataforma Android, quando ela surgiu e todos os componentes que a permeiam, além disso, buscará apresentar o que é o desenvolvimento de um aplicativo, conceituando o que é a linguagem Java, o que é um framework e apresentar o Eclipse Modeling Framework, por fim, buscará apresentar e detalhar as principais etapas voltadas para a proposta do desenvolvimento de um aplicativo para academias. Por fim, o presente trabalho deixa o tema em aberto, propondo que no futuro se realize uma nova pesquisa, com a finalidade de contextualizar os temas aqui abordados. Juntamente com esta nova pesquisa, sugere-se a realização de um novo estudo de caso, para o qual propõe-se uma análise de mercado quanto a aceitação de tal aplicativo. Juntamente com esta análise de mercado sugere-se que se valide a proposta aqui apresentada e aplique uma análise de campo prática, com a finalidade de verificar sua efetivação.

Palavras-chave: Desenvolvimento; Aplicativo; Linguagem Java.

SUMÁRIO

1 INTRODUÇÃO	2
1.1 PROBLEMA	4
1.2 JUSTIFICATIVA	5
1.3 OBJETIVOS	5
1.3.1 Objetivo Geral	5
1.3.2 Objetivos Específicos	5
2 REVISÃO DA LITERATURA	6
2.1 DISPOSITIVOS MÓVEIS	6
2.2 ANDROID	8
2.3 DESENVOLVIMENTO DE APLICATIVOS	10
2.3.1 Linguagem Java	12
2.3.2 Framework	14
2.3.3 Eclipse Modeling Framework (EMF)	14
2.4 DESENVOLVIMENTO DE APLICATIVOS	15
2.4.1 Linguagem Java	17
2.4.2 Framework	19
2.4.3 Eclipse Modeling Framework (EMF)	20
3 PROCEDIMENTOS METODOLÓGICOS	21
3.1 CARACTERIZAÇÃO DO ESTUDO	21
3.2 CAMPO DE ESTUDO	23
3.3 INSTRUMENTOS PARA COLETA DE DADOS	23
4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS	24
5 CONSIDERAÇÕES FINAIS	33
REFERÊNCIAS	34
ANEXOS	37

1 INTRODUÇÃO

A importância dos telefones celulares na sociedade e em suas atividades cotidianas é inegavelmente interminável. Isso ocorre porque há uma tremenda transformação em andamento, pois os telefones celulares não são mais o dispositivo de comunicação comum que costumava ser. Tornou-se o ponto de atenção colossal para indivíduos e empresas, cortesia dos vários recursos e oportunidades incríveis que os telefones celulares oferecem. O progresso cumulativo da tecnologia móvel, a disponibilidade e acesso à Internet de alta velocidade e a notável interface comunicativa nesses dispositivos resultam em um nível inteiro de experiência móvel nova e inovadora na computação móvel. Isso é possível através do desenvolvimento de aplicativos móveis (DEHLING, et al. 2015).

Hoje, a disponibilidade de aplicativos móveis está aumentando, de modo a produzir uma mudança notável na maneira como os seres humanos sentem e experimentam a computação. Alguns anos atrás, para acessar a Internet, verificar e ler e-mails, era necessário usar o computador, mas hoje isso mudou porque a computação agora é transportada para todos os lugares nos telefones celulares. Imagine comprar uma passagem de trem em movimento, isso é algo que nossos ancestrais nunca imaginaram ou fizeram (EL-KASSAS, 2017).

Imagine não ir ao banco, mas ainda transferir dinheiro para familiares e amigos. Tudo graças aos desenvolvedores de aplicativos e às principais empresas de desenvolvimento de aplicativos. Não importa qual, eles vieram para o resgate, permitindo uma vida fácil (BJORN-HANSEN, et al. 2018).

Em geral, a presença de aplicativos móveis em telefones pode ser comparada às decorações de um bolo, pois tornam os celulares agradáveis e divertidos. Os aplicativos móveis são ilimitados em número, com o uso que abrange todas as esferas da vida e com as pessoas que desejam mais e mais desses aplicativos para facilitar o estilo de vida e a vida. Atualmente, o uso de aplicativos móveis pode ser visto em áreas como comunicação, educação, culinária, mídias sociais, compras, negócios (geração de dinheiro), casamento e serviços bancários (GRANDGEORGE, et al. 2016).

Diariamente, os indivíduos procuram versões atualizadas destes aplicativos. Portanto, desenvolvedores de aplicativos e empresas de desenvolvimento de aplicativos estão sempre trabalhando para manter-se atualizado com as demandas. Devido a isso, houve um aumento recente no crescimento de novos desenvolvedores de aplicativos e empresas de desenvolvimento de aplicativos (EL-KASSAS, 2017).

Hoje, o telefone celular se tornou parte integrante de tantas pessoas devido a esses aplicativos, de modo que se poderia dizer que ajuda a manter algum tipo de vida organizada. A disponibilidade de aplicativos para contatos, projetos e eventos relevantes, informações pessoais e eventos futuros em telefones celulares atesta isso (DESHPANDE, et al. 2017).

Esse conjunto de informações é totalmente armazenado em nossos telefones e pode nos ajudar a planejar a vida, facilitando assim o gerenciamento adequado do tempo. Cortesia de vários aplicativos, como alarmes, lembretes, listas de tarefas e todos os tipos de aplicativos de notificação que podem ser programados de acordo com nossas necessidades e exigências individuais, tornando a vida mais confortável, mais fácil e mais produtiva (GRANDGEORGE, et al. 2016).

Neste contexto esta pesquisa tem como tema, apresentar uma proposta de aplicativo para academias para dispositivos android. A indústria de ginástica atende uma pequena parte da população, se considerar que o Brasil tem quase 200 milhões de habitantes. No entanto, esse setor vem ganhando espaço e investimentos, principalmente porque as pessoas estão cada vez mais conscientes da importância da prática de exercícios para ter uma vida mais saudável (MELLO, 2013).

De acordo com uma pesquisa feita pelo Instituto Data Popular, 79 milhões de brasileiros declararam ter interesse em praticar qualquer atividade física, sendo 54,2% deles de C classe social . Essa classe é responsável por boa parte da alta do mercado; agora que sua renda é maior, eles podem bancar academias - algo que antes era um luxo -, visando uma melhor qualidade de vida.

A grande dificuldade da maioria das academias de ginástica do Brasil consiste em atendimento. Devido ao aumento acelerado pela procura de milhares de pessoas por praticar exercícios, as academias de ginástica atendem por mais de 12 horas diariamente, necessitando assim de um grande quadro de funcionários para instruir seus alunos. O que gera grandes encargos financeiros e não apresenta um bom resultado no atendimento.

A indústria brasileira de academias é a segunda maior do mundo. Pela terceira vez consecutiva, o Brasil ocupa a segunda posição na lista dos países com maior número de academias do mundo. A preocupação com a saúde e a busca pelo corpo perfeito colocam o Brasil entre as 3 primeiras nações, com mais de 23,6 mil estabelecimentos abertos no país, atrás apenas das 30 mil academias dos Estados Unidos (MELLO, et al. 2013).

De acordo com o Relatório Latino-Americano 2012, feito pela International Health, Racquet and Sportsclub Association (IHRSA), somente em 2011, o número de academias cresceu 30% e o faturamento do setor de fitness e bem-estar aumentou 10%, chegando a R \$

2,45 bilhões. Com essa receita, o Brasil ocupa a terceira posição no ranking da indústria de ginástica nas Américas, atrás dos Estados Unidos e Canadá.

A associação brasileira de academias de ginástica denominada "Associação Brasileira de Academias", também conhecida como ACAD, afirma que mais de 5 milhões de brasileiros - o que corresponde a 3% da população do país - frequentam academias. Dizem também que esse número pode aumentar ainda mais, tendo em vista que nações como Holanda e Dinamarca têm 40% de sua população como integrantes desses estabelecimentos (DECKERT, 2014).

São Paulo é o estado com o maior número de academias de ginástica do Brasil: mais de 6.400 estabelecimentos, uma para cada 6.445 habitantes. Santa Catarina é o estado com a segunda maior densidade de academias; mais de 1.700 deles, um para cada 3.487 pessoas.

Aproximadamente 30% das pessoas que frequentam academias de ginástica no Brasil têm mais de 60 anos. Há uma década, esse número era estimado em apenas 5%. A terceira idade contribui para a ascensão do setor, pois costumam ser mais fiéis às academias. Mais do que isso, entendem que a prática de exercícios físicos é importante para uma melhor qualidade de vida (LIMA, 2013).

Os exercícios físicos, junto com outros benefícios como o fortalecimento muscular, ajudam no controle do diabetes, artrite e doenças cardíacas, além de melhorar os reflexos e a flexibilidade do corpo. Também é importante para controlar a pressão arterial (DECKERT, 2014).

Quando o indivíduo se exercita, seu corpo estimula a produção de serotonina, uma substância química do cérebro que transporta mensagens por todo o sistema nervoso. A serotonina regula o apetite, o sono e o humor. A prática de exercícios proporciona às pessoas um melhor humor e evita que tenham depressão. Além disso, se os exercícios forem feitos em grupo, ajuda na socialização - o que é tão importante quanto qualquer outro benefício (LIMA, 2013).

1.1 PROBLEMA

Desta maneira, o presente trabalho buscará responder como o desenvolvimento de uma proposta de aplicativo para aparelhos Android, pode vir a ser um diferencial comercial ao ser aplicado em uma academia?

1.2 JUSTIFICATIVA

Justifica-se a motivação desta pesquisa como um meio de abordar um assunto tão pertinente com o cotidiano das pessoas. Apresentar a relação da tecnologia como um facilitador de tarefas, ferramentas e de atividades físicas como aqui abordado é algo totalmente relevante, gerando assim uma motivação empreendedora e social, tendo em vista a contribuição positiva que tal projeto pode gerar na sociedade.

Justifica-se, também como um meio de contribuir para o ambiente acadêmico, contextualizando e enriquecendo a temática referente a proposta do desenvolvimento de um aplicativo android para academias.

Por fim, justifica-se esta pesquisa, como um meio de simplificar tal temática em seu ambiente social, buscando apresentar um material conciso e de fácil assimilação por leitores leigos que buscam um conhecimento mais profundo no tema.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

O objetivo geral desta pesquisa, buscará apresentar como o desenvolvimento de uma proposta de aplicativo para aparelhos Android, pode vir a ser um diferencial comercial ao ser aplicado em uma academia

1.3.2 Objetivos Específicos

Os objetivos específicos do presente trabalho buscarão:

1. Apresentar e conceituar o que é um dispositivo móvel;
2. Salientar o que é a plataforma Android, quando ela surgiu e todos os componentes que a permeiam;
3. Apresentar o que é o desenvolvimento de um aplicativo, conceituando o que é a linguagem Java, o que é um framework e apresentar o Eclipse Modeling Framework;
4. Apresentar e detalhar as principais etapas voltadas para a proposta do desenvolvimento de um aplicativo como diferencial comercial para academias.

2 REVISÃO DA LITERATURA

2.1 DISPOSITIVOS MÓVEIS

Um dispositivo móvel é um termo geral para qualquer tipo de computador portátil. Esses dispositivos foram projetados para serem extremamente portáteis e geralmente podem caber na sua mão. Alguns dispositivos móveis - como tablets, leitores eletrônicos e smartphones, são poderosos o suficiente para fazer as mesmas coisas que você pode fazer com um desktop ou laptop (DEHLING, et al. 2015).

Muitos desses dispositivos podem se conectar à Internet e interligar-se com outros dispositivos, como sistemas de entretenimento automóvel ou fones de ouvido via Wi-Fi, Bluetooth, redes celulares ou *Near Field Communication* (NFC). Câmeras integradas, capacidade de colocar e receber telefone de voz e videochamadas, videogames e recursos do Sistema de Posicionamento Global (GPS) são comuns. Normalmente, a energia é fornecida por uma bateria de lítio. Os dispositivos móveis podem executar sistemas operacionais móveis que permitem que aplicativos de terceiros especializados para esses recursos sejam instalados e executados (OGLIARI, et al. 2015).

No início dos anos 2000, os primeiros smartphones foram juntados por tablets maiores, mas essencialmente iguais. A entrada e a saída agora são normalmente via interface *touch-screen*. Telefones / tablets e assistentes digitais pessoais podem fornecer grande parte da funcionalidade de um laptop / computador de mesa, mas de forma mais conveniente, além de recursos exclusivos. Os assistentes digitais corporativos podem fornecer funcionalidades comerciais adicionais, como captura de dados integrada por código de barras, RFID e cartão inteligente. Em 2010, os dispositivos móveis geralmente continham sensores como acelerômetros, magnetômetros e giroscópios, permitindo a detecção de orientação e movimento. Os dispositivos móveis podem fornecer autenticação biométrica do usuário, como reconhecimento facial ou reconhecimento de impressão digital (LUPTON, et al. 2015).

Segundo Baig (2015), a mobilidade do dispositivo pode ser visualizada no contexto de várias dimensões:

1. Dimensões físicas e peso
2. Se o dispositivo é ou não móvel ou algum tipo de host ao qual está conectado é móvel
3. A que tipo de dispositivos host pode ser vinculado
4. Como os dispositivos são conectados a um host

5. Quando a mobilidade ocorre

A rigor, muitos dos chamados dispositivos móveis não são móveis. É o *host* móvel, ou seja, um *host* humano móvel carrega um dispositivo de smartphone não móvel. Um exemplo de um verdadeiro dispositivo de computação móvel, onde o próprio dispositivo é móvel, é um robô. Outro exemplo é um veículo autônomo (SILVA, et al. 2015).

Há três maneiras básicas de os dispositivos móveis serem fisicamente vinculados aos hosts móveis: acompanhados, montados na superfície ou incorporados na estrutura de um *host*, por exemplo, um controlador incorporado, incorporado em um dispositivo host (BAIG, et al. 2015).

Acompanhado refere-se a um objeto que está vagamente preso e acompanha um host móvel, por exemplo, um smartphone pode ser carregado em uma bolsa ou bolso, mas pode ser facilmente extraviado. Portanto, hosts móveis com dispositivos incorporados, como um veículo autônomo pode parecer maior que o tamanho de um bolso (GRANDGEORGE, et al. 2016).

Como afirmado anteriormente, o tamanho mais comum de dispositivo de computação móvel é do tamanho de um bolso que pode ser portátil, mas também existem outros tamanhos para dispositivos móveis. Mark Weiser, conhecido como o pai da computação onipresente, computando em todos os lugares, refere-se a tamanhos de dispositivos com tamanho de tabulação, tamanho de bloco e placa, onde guias são definidas como dispositivos de tamanho centímetro acompanhados ou vestíveis, por exemplo, smartphones e blocos são definidos como dispositivos portáteis do tamanho de decímetros, por exemplo, *laptops*. Se alguém muda a forma dos dispositivos móveis em termos de não ser plano, também pode haver dispositivos de capa e pequenos dispositivos do tamanho de poeira. Poeira refere-se a dispositivos miniaturizados sem interfaces HCI diretas, por exemplo, sistemas micro eletromecânicos (MEMS), variando de nanômetros a micrômetros e milímetros (PATRICK, 2015).

Embora a mobilidade seja frequentemente considerada como sinônimo de conectividade sem fio, esses termos são diferentes. Nem todo acesso à rede por usuários móveis, aplicativos e dispositivos precisa ser via redes sem fio e vice-versa. Os dispositivos de acesso sem fio podem ser estáticos e os usuários móveis podem se mover entre pontos de acesso com e sem fio, como em cybercafés. Alguns dispositivos móveis podem ser usados como dispositivos móveis da Internet para acessar a Internet em movimento, mas eles não precisam fazer isso e muitas funções ou aplicativos do telefone ainda estão operacionais, mesmo quando desconectados da Internet (DESHPANDE, et a. 2017).

O que torna o dispositivo móvel único em comparação com outras tecnologias é a flexibilidade inerente ao *hardware* e também ao software. Aplicações flexíveis incluem bate-papo por vídeo, navegação na *Web*, sistemas de pagamento, NFC, gravação de áudio etc. À medida que os dispositivos móveis se tornarem onipresentes lá, haverá uma proliferação de serviços que incluem o uso da nuvem (OGLIARI, et al. 2015).

Embora uma forma comum de dispositivo móvel, um smartphone, tenha uma tela, outra forma talvez ainda mais comum de dispositivo de computação inteligente, o cartão inteligente, por exemplo, usado como cartão bancário ou cartão de viagem, não possui visor. Esse dispositivo móvel geralmente possui uma CPU e memória, mas precisa se conectar ou ser inserido em um leitor para exibir seus dados ou estado interno (MULLEN, et al. 2017).

2.2 ANDROID

O Android é um sistema operacional baseado em Linux, projetado principalmente para dispositivos móveis com tela de toque, como smartphones e tablets. O sistema operacional se desenvolveu muito nos últimos 15 anos, começando em telefones preto e branco e smartphones ou mini computadores recentes. Um dos sistemas operacionais móveis mais utilizados atualmente é o Android. O android é um software que foi fundado em Palo Alto, na Califórnia, em 2003 (ANDROID, 2019).

O android é um poderoso sistema operacional e suporta grande número de aplicativos em smartphones. Esses aplicativos são mais confortáveis e avançados para os usuários. O *hardware* que suporta o *software* Android é baseado na plataforma de arquitetura ARM. O android é um sistema operacional de código aberto, que é gratuito e qualquer um pode usá-lo. O Android tem milhões de aplicativos disponíveis que podem ajudá-lo a gerenciar sua vida de uma maneira ou de outra e está disponível no mercado com baixo custo, por isso o Android é muito popular (BJORN-HANSEN, et al. 2018).

O desenvolvimento do Android é compatível com a linguagem de programação java completa. Mesmo outros pacotes que são API e JSE não são suportados. A primeira versão 1.0 do kit de desenvolvimento do Android (SDK) foi lançada em 2008 e a versão mais recente atualizada é o Jelly Bean. A arquitetura do Android, é baseada em um sistema operacional e uma pilha de componentes de software divididos em cinco seções e quatro camadas principais (HARRISON, 2015):

- 2 Bibliotecas
- 3 Tempo de execução do Android

O Android usa o poderoso kernel Linux e suporta uma ampla gama de drivers de hardware. O kernel é o coração do sistema operacional que gerencia solicitações de entrada e saída de *software*. Isso fornece funcionalidades básicas do sistema, como gerenciamento de processos, gerenciamento de memória, gerenciamento de dispositivos como câmera, teclado, tela etc (SARTOR, et al. 2015).

O kernel lida com tudo. O Linux é realmente bom em redes e não é necessário fazer interface com o hardware periférico. O próprio kernel não interage diretamente com o usuário, mas interage com o shell e outros programas, bem como com os dispositivos de hardware do sistema (GARGENTA, 2015).

No topo de um kernel Linux, há um conjunto de bibliotecas, incluindo navegador de código aberto, como *webkit*, *library libc*. Essas bibliotecas são usadas para reproduzir e gravar áudio e vídeo. O *SQLite* é uma base de dados útil para armazenamento e compartilhamento de dados de aplicativos. As bibliotecas SSL são responsáveis pela segurança da Internet, etc (EL-KASSAS, 2017).

O tempo de execução do Android fornece um componente-chave chamado *Dalvik Virtual Machine*, que é um tipo de máquina virtual java. É especialmente projetado e otimizado para Android. A *Dalvik VM* é a máquina virtual de processo no sistema operacional Android. É um software que executa aplicativos em dispositivos Android (TANENBAUM, 2015).

A *Dalvik VM* faz uso dos principais recursos do Linux, como gerenciamento de memória e multithreading, que estão em uma linguagem java. A *Dalvik VM* permite que cada aplicativo Android execute seu próprio processo. A *Dalvik VM* executa os arquivos no formato *.dex*. A camada de trabalho do quadro de aplicativos fornece muitos serviços de nível superior para aplicativos como gerenciador de janelas, sistema de exibição, gerenciador de pacotes, gerenciador de recursos etc. Os desenvolvedores de aplicativos podem fazer uso desses serviços em seus aplicativos (BEZERRA, et al. 2016).

O código-fonte do Android foi usado como base de diferentes ecossistemas, principalmente o do Google, associado a um conjunto de software proprietário chamado *Google Mobile Services (GMS)*, que frequentemente vem pré-instalado nos referidos dispositivos. Isso inclui aplicativos principais, como o Gmail, a plataforma de distribuição digital *Google Play* e a plataforma de desenvolvimento associada ao *Google Play Services*, e

geralmente aplicativos como o navegador Google Chrome. Esses aplicativos são licenciados por fabricantes de dispositivos Android certificados sob os padrões impostos pelo Google (LECHETA, 2015).

Outros ecossistemas Android concorrentes incluem *Amazon.com 'sFire OS* ou *LineageOS*. A distribuição de software geralmente é oferecida por lojas de aplicativos proprietários, como *Google Play Store* ou *Samsung Galaxy Store*, ou plataformas de código aberto, como *Aptoide* ou *F-Droid*, que usam pacotes de software no formato *APK*. O Android é o sistema operacional mais vendido em todo o mundo em smartphones desde 2011 e tablets desde 2013. Em maio de 2017, conta com mais de dois bilhões de usuários ativos mensais, a maior base instalada de qualquer sistema operacional (BJORN-HANSEN, et al. 2018).

Em maio de 2019, o sistema operacional se envolveu na guerra comercial entre a China e os Estados Unidos envolvendo a *Huawei*, que, como muitas outras empresas de tecnologia, se tornou dependente do acesso à plataforma Android. No verão de 2019, a *Huawei* anunciou que criaria um sistema operacional alternativo ao Android, conhecido como *Harmony OS*, e solicitou direitos de propriedade intelectual nos principais mercados globais. Atualmente, a *Huawei* não tem planos de substituir o Android em um futuro próximo, pois o *Harmony OS* foi projetado para a Internet das coisas dispositivos, e não para smartphones (ANDROID, 2019).

Em 22 de agosto de 2019, foi anunciado que o Android "Q" seria oficialmente marcado como Android 10, encerrando a prática histórica de nomear versões principais após sobremesas. O Google afirmou que esses nomes não eram "inclusivos" para usuários internacionais (devido ao fato de os alimentos acima mencionados não serem conhecidos internacionalmente ou por serem difíceis de pronunciar em alguns idiomas). No mesmo dia, a *Android Police* informou que o Google encomendou uma estátua de um número gigante "10" para ser instalada no *lobby* do novo escritório dos desenvolvedores. O Android 10 foi lançado em 3 de setembro de 2019 para os telefones Google Pixel primeiro (ANDROID, 2019).

2.3 DESENVOLVIMENTO DE APLICATIVOS

Desenvolvimento de aplicativo móvel é o ato ou processo pelo qual um aplicativo móvel é desenvolvido para dispositivos móveis, como assistentes digitais pessoais, assistentes digitais corporativos ou telefones celulares. Esses aplicativos podem ser pré-instalados nos telefones durante as plataformas de fabricação ou entregues como aplicativos da Web usando

o processamento no servidor ou no cliente (por exemplo, JavaScript) para fornecer uma experiência "semelhante ao aplicativo" em um navegador da Web (SARTOR, et al. 2015).

Os desenvolvedores de software de aplicativo também devem considerar uma grande variedade de tamanhos de tela, especificações de hardware e configurações devido à intensa concorrência no software móvel e às mudanças em cada uma das plataformas. O desenvolvimento de aplicativos para dispositivos móveis tem crescido constantemente, em receitas e empregos criados. Um relatório de analistas de 2013 estima que existam 529.000 empregos diretos na economia de aplicativos na UE e 28 membros (incluindo o Reino Unido), 60% dos quais são desenvolvedores de aplicativos móveis (GARGENTA, et al. 2015).

Como parte do processo de desenvolvimento, a interface de usuário móvel. O design (UI) também é essencial na criação de aplicativos móveis. A interface do usuário móvel considera restrições, contextos, tela, entrada e mobilidade como contornos para o design. O usuário geralmente é o foco da interação com o dispositivo, e a interface envolve componentes de *hardware* e *software*. A entrada do usuário permite que os usuários manipulem um sistema, e a saída do dispositivo permite que o sistema indique os efeitos da manipulação dos usuários (CLARK, et al. 2016).

As restrições de design da interface do usuário móvel incluem atenção e fatores de forma limitados, como o tamanho da tela de um dispositivo móvel para as mãos de um usuário. Os contextos da interface do usuário para dispositivos móveis sinalizam dicas da atividade do usuário, como localização e programação que podem ser mostradas nas interações do usuário em um aplicativo móvel (AMADEO, 2017).

No geral, o objetivo do design de interface do usuário móvel é principalmente para uma interface compreensível e amigável. A interface do usuário dos aplicativos móveis deve: considere a atenção limitada dos usuários, minimize as teclas digitadas e seja orientado a tarefas com um conjunto mínimo de funções. Essa funcionalidade é suportada por plataformas de aplicativos corporativos móveis ou ambientes de desenvolvimento integrado (IDEs) (BOHN, 2019).

As UIs para dispositivos móveis, ou *front-ends*, contam com *back-ends* móveis para dar suporte ao acesso aos sistemas corporativos. O *back-end* móvel facilita o roteamento de dados, segurança, autenticação, autorização, trabalho off-line e orquestração de serviços (BEZERRA, et al. 2016). Essa funcionalidade é suportada por uma mistura de componentes de *middleware*, incluindo servidor de aplicativos móveis, *back-end* móvel como serviço (MBaaS) e infraestrutura de arquitetura orientada a serviços (SOA) (EADICICCO, 2015).

O desenvolvimento de aplicativos móveis está se tornando mais crítico para muitas empresas, com mais de 3 bilhões de pessoas em todo o mundo usando smartphones, mais de 1,5 bilhão usando tablets a partir de 2019. Os usuários, em média, gastam 90% de seu tempo móvel em aplicativos e há mais de 700 milhões de *downloads* de aplicativos de várias lojas de aplicativos (LECHETA, 2015).

2.3.1 Linguagem Java

O Java é uma de uso geral linguagem de programação que é baseada em classe, objeto orientado, e projetado para ter o menor número de implementação dependências quanto possíveis. Ele tem como objetivo permitir que os desenvolvedores de aplicativos escrevam uma vez, executem em qualquer lugar (WORA), o que significa que o código Java compilado pode ser executado em todas as plataformas que suportam Java sem a necessidade de recompilação (TANENBAUM, 2015).

Os aplicativos Java geralmente são compilados no *bytecode* que pode ser executado em qualquer *Java Virtual Machine* (JVM), independentemente da arquitetura do computador subjacente. A sintaxe do Java é semelhante ao C e C ++, mas possui menos recursos de baixo nível do que qualquer um deles. A partir de 2019, Java era uma das linguagens de programação mais populares em uso, de acordo com o GitHub, particularmente para aplicativos Web cliente-servidor, com 9 milhões de desenvolvedores relatados (EL-KASSAS, 2017).

O Java foi originalmente desenvolvido por James Gosling na *Sun Microsystems* (que desde então foi adquirida pela Oracle) e lançado em 1995 como um componente principal da plataforma Java da *Sun Microsystems*. Os compiladores Java, máquinas virtuais e bibliotecas de classes de implementação original e de referência foram originalmente lançados pela *Sun* sob licenças proprietárias (LUCKOW, 2015).

Em maio de 2007, em conformidade com as especificações do *Java Community Process*, a *Sun* havia licenciado a maioria de suas tecnologias Java sob a Licença Pública Geral GNU. Enquanto isso, outros desenvolveram implementações alternativas dessas tecnologias da Sun, como o *GNU Compiler for Java* (compilador de *bytecode*), *GNU Classpath* (bibliotecas padrão) e *IcedTea -Web* (plug-in de navegador para *applets*) (GOSLING, 2015).

As versões mais recentes são Java 13, lançado em setembro de 2019, e Java 11, uma versão de suporte a longo prazo (LTS) atualmente suportada, lançada em 25 de setembro de

2018; A Oracle lançou para o Java 8 LTS legado a última atualização pública gratuita em janeiro de 2019 para uso comercial, enquanto ainda suportará o Java 8 com atualizações públicas para uso pessoal até pelo menos dezembro de 2020. A *Oracle* (e outros) recomendam altamente a desinstalação antiga versões do Java devido a sérios riscos devido a problemas de segurança não resolvidos. Como o Java 9 (e 10 e 12) não é mais suportado, a *Oracle* aconselha seus usuários a migrarem imediatamente para a versão mais recente (atualmente Java 13) ou uma versão LTS (SILVA, 2015).

A linguagem Java é um pilar fundamental do Android, um sistema operacional móvel de código aberto. Embora o Android, construído no kernel do Linux, seja escrito principalmente em C, o Android SDK usa a linguagem Java como base para aplicativos Android, mas não usa nenhum de seus GUI, SE, ME ou outros padrões Java estabelecidos. A linguagem de *bytecode* suportada pelo Android SDK é incompatível com o *bytecode* Java e é executada em sua própria máquina virtual, otimizada para dispositivos com pouca memória, como smartphones e tablets. Dependendo da versão do Android, o *bytecode* é interpretado pela máquina virtual *Dalvikou* compilado em código nativo pelo *Android Runtime* (LUCKOW, 2015).

O Android não fornece a biblioteca padrão completa do Java SE, embora o Android SDK inclua uma implementação independente de um grande subconjunto dela. Ele suporta Java 6 e alguns recursos do Java 7, oferecendo uma implementação compatível com a biblioteca padrão (*Apache Harmony*).

O uso da tecnologia relacionada ao Java no Android levou a uma disputa legal entre Oracle e Google. Em 7 de maio de 2012, um júri de São Francisco descobriu que, se as APIs pudessem ter direitos autorais, o Google violaria os direitos autorais da Oracle pelo uso de Java em dispositivos Android. O juiz distrital William Haskell Alsup decidiu em 31 de maio de 2012 que APIs não podem ser protegidas por direitos autorais, mas isso foi revertido pelo Tribunal de Apelações dos Estados Unidos para o Circuito Federal em maio de 2014 (GOSLING, 2015).

Em 26 de maio, 2016, o tribunal distrital decidiu a favor do Google, declarando que a violação de direitos autorais da API Java no Android constitui uso justo. Em março de 2018, essa decisão foi revogada pelo Tribunal de Apelações, que enviou o caso de determinação dos danos ao tribunal federal de São Francisco. O Google entrou com uma petição de ordem de *certiorari* no Supremo Tribunal dos Estados Unidos em janeiro de 2019 para contestar as duas decisões que foram tomadas pelo Tribunal de Apelações em favor da Oracle (BOHN, 2019).

2.3.2 Framework

No desenvolvimento de aplicativos, desenvolvimento de *software* e desenvolvimento *web*, as estruturas são bibliotecas de códigos com funções comuns definidas. Dessa forma, o usuário precisa apenas de uma pequena linha de código para realizar o que de outra forma precisaria de longos trechos de código. Isso ajuda a acelerar o processo de desenvolvimento para programadores (SILVA, 2015).

Uma maneira de visualizar uma estrutura é imaginar casas pré-fabricadas. Os diferentes módulos são pré-construídos. Eles são simplesmente reunidos no site. Você não está 'construindo' uma porta do zero. Você está apenas pegando uma porta pré-montada e encaixando-a. Isso economiza tempo (MULLEN, et al. 2017).

Um bom exemplo de estrutura seria *o.NET Core*, uma estrutura de código aberto da Microsoft. Essa estrutura é ótima para criar aplicativos para os sistemas operacionais comuns - Windows, Mac OS e Linux - desde que você esteja confortável trabalhando com C # (OGLIARI, et al. 2015).

As estruturas de aplicativos tornaram-se populares com o aumento das interfaces gráficas de usuário (GUIs), uma vez que elas tendiam a promover uma estrutura padrão para aplicativos. Os programadores acham muito mais simples criar ferramentas automáticas de criação de GUI ao usar uma estrutura padrão, pois isso define a estrutura de código subjacente do aplicativo com antecedência. Os desenvolvedores geralmente usam técnicas de programação orientada a objetos (OOP) para implementar estruturas de modo que as partes exclusivas de um aplicativo possam simplesmente herdar das classes existentes na estrutura (DESHPANDE, et al. 2017).

2.3.3 Eclipse Modeling Framework (EMF)

O *Eclipse Modeling Framework* (EMF) é uma estrutura de modelagem baseada em Eclipse e um recurso de geração de código para a construção de ferramentas e outros aplicativos baseados em um modelo de dados estruturado (DESHPANDE, et al. 2017).

Em outras palavras, o projeto EMF é uma estrutura de modelagem e um recurso de geração de código para a construção de ferramentas e outros aplicativos com base em um modelo de dados estruturado. A partir de uma especificação de modelo descrita em XMI, o EMF fornece ferramentas e suporte de tempo de execução para produzir um conjunto de classes Java para o modelo, juntamente com um conjunto de classes de adaptadores que

permitem a visualização e edição baseada em comando do modelo e um editor básico (GRANDGEORGE, et al. 2016).

O EMF (núcleo) é um padrão comum para modelos de dados, com base em muitas tecnologias e estruturas. Isso inclui soluções de servidor, estruturas de persistência, estruturas de interface do usuário e suporte para transformações. Dê uma olhada no projeto de modelagem para obter uma visão geral das tecnologias EMF (SILVA, et al. 2015).

A partir de uma especificação de modelo descrita em XML *Metadata Interchange* (XMI), o EMF fornece ferramentas e suporte de tempo de execução para produzir um conjunto de classes Java para o modelo, um conjunto de classes de adaptadores que permitem a visualização e edição baseada em comando do modelo e uma base editor. Os modelos podem ser especificados usando documentos Java, UML, XML ou ferramentas de modelagem anotados e importados para o EMF. Mais importante de tudo, o EMF fornece a base para a interoperabilidade com outras ferramentas e aplicativos baseados no EMF (BAIG, et al. 2015).

O *Ecore* é o (meta) modelo central no coração da EMF. Permite expressar outros modelos, alavancando suas construções. O *Ecore* também é seu próprio metamodelo (ou seja: o *Ecore* é definido em termos de si mesmo) (PATRICK, et al. 2015).

De acordo com *Ed Merks*, líder do projeto EMF, "*Ecore* é a implementação de referência padrão do EMOF da OMG " (*Essential Meta-Object Facility*). Ainda de acordo com *Merks*, o EMOF foi realmente definido pelo OMG como uma versão simplificada do 'C'MOF mais abrangente, aproveitando a experiência da simplificação bem-sucedida da implementação original do *Ecore* (BAIG, et al. 2015).

O uso do *Ecore* como um metamodelo de base permite que um modelador aproveite todo o ecossistema e ferramentas EMF - na medida em que seja razoavelmente fácil mapear modelos no nível de aplicativo de volta ao *Ecore*. Isso não quer dizer que é uma prática recomendada os aplicativos alavancarem diretamente o *Ecore* como seu metamodelo; em vez disso, eles podem considerar definir seus próprios metamodelos com base no *Ecore* (SILVA, et al. 2015).

2.4 DESENVOLVIMENTO DE APLICATIVOS

Desenvolvimento de aplicativo móvel é o ato ou processo pelo qual um aplicativo móvel é desenvolvido para dispositivos móveis, como assistentes digitais pessoais, assistentes

digitais corporativos ou telefones celulares. Esses aplicativos podem ser pré-instalados nos telefones durante as plataformas de fabricação ou entregues como aplicativos da Web usando o processamento no servidor ou no cliente (por exemplo, JavaScript) para fornecer uma experiência "semelhante ao aplicativo" em um navegador da Web (SARTOR, et al. 2015).

Os desenvolvedores de software de aplicativo também devem considerar uma grande variedade de tamanhos de tela, especificações de hardware e configurações devido à intensa concorrência no software móvel e às mudanças em cada uma das plataformas. O desenvolvimento de aplicativos para dispositivos móveis tem crescido constantemente, em receitas e empregos criados. Um relatório de analistas de 2013 estima que existam 529.000 empregos diretos na economia de aplicativos na UE e 28 membros (incluindo o Reino Unido), 60% dos quais são desenvolvedores de aplicativos móveis (GARGENTA, et al. 2015).

Como parte do processo de desenvolvimento, a interface de usuário móvel. O design (UI) também é essencial na criação de aplicativos móveis. A interface do usuário móvel considera restrições, contextos, tela, entrada e mobilidade como contornos para o design. O usuário geralmente é o foco da interação com o dispositivo, e a interface envolve componentes de *hardware* e *software*. A entrada do usuário permite que os usuários manipulem um sistema, e a saída do dispositivo permite que o sistema indique os efeitos da manipulação dos usuários (CLARK, et al. 2016).

As restrições de design da interface do usuário móvel incluem atenção e fatores de forma limitados, como o tamanho da tela de um dispositivo móvel para as mãos de um usuário. Os contextos da interface do usuário para dispositivos móveis sinalizam dicas da atividade do usuário, como localização e programação que podem ser mostradas nas interações do usuário em um aplicativo móvel (AMADEO, 2017).

No geral, o objetivo do design de interface do usuário móvel é principalmente para uma interface compreensível e amigável. A interface do usuário dos aplicativos móveis deve: considere a atenção limitada dos usuários, minimize as teclas digitadas e seja orientado a tarefas com um conjunto mínimo de funções. Essa funcionalidade é suportada por plataformas de aplicativos corporativos móveis ou ambientes de desenvolvimento integrado (IDEs) (BOHN, 2019).

As UIs para dispositivos móveis, ou *front-ends*, contam com *back-ends* móveis para dar suporte ao acesso aos sistemas corporativos. O *back-end* móvel facilita o roteamento de

dados, segurança, autenticação, autorização, trabalho off-line e orquestração de serviços (BEZERRA, et al. 2016). Essa funcionalidade é suportada por uma mistura de componentes de *middleware*, incluindo servidor de aplicativos móveis, *back-end* móvel como serviço (MBaaS) e infraestrutura de arquitetura orientada a serviços (SOA) (EADICICCO, 2015).

O desenvolvimento de aplicativos móveis está se tornando mais crítico para muitas empresas, com mais de 3 bilhões de pessoas em todo o mundo usando smartphones, mais de 1,5 bilhão usando tablets a partir de 2019. Os usuários, em média, gastam 90% de seu tempo móvel em aplicativos e há mais de 700 milhões de *downloads* de aplicativos de várias lojas de aplicativos (LECHETA, 2015).

2.4.1 Linguagem Java

O Java é uma de uso geral linguagem de programação que é baseada em classe, objeto orientado, e projetado para ter o menor número de implementação dependências quanto possíveis. Ele tem como objetivo permitir que os desenvolvedores de aplicativos escrevam uma vez, executem em qualquer lugar (WORA), o que significa que o código Java compilado pode ser executado em todas as plataformas que suportam Java sem a necessidade de recompilação (TANENBAUM, 2015).

Os aplicativos Java geralmente são compilados no *bytecode* que pode ser executado em qualquer *Java Virtual Machine* (JVM), independentemente da arquitetura do computador subjacente. A sintaxe do Java é semelhante ao C e C ++, mas possui menos recursos de baixo nível do que qualquer um deles. A partir de 2019, Java era uma das linguagens de programação mais populares em uso, de acordo com o GitHub, particularmente para aplicativos Web cliente-servidor, com 9 milhões de desenvolvedores relatados (EL-KASSAS, 2017).

O Java foi originalmente desenvolvido por James Gosling na *Sun Microsystems* (que desde então foi adquirida pela Oracle) e lançado em 1995 como um componente principal da plataforma Java da *Sun Microsystems*. Os compiladores Java, máquinas virtuais e bibliotecas

de classes de implementação original e de referência foram originalmente lançados pela *Sun* sob licenças proprietárias (LUCKOW, 2015).

Em maio de 2007, em conformidade com as especificações do *Java Community Process*, a *Sun* havia licenciado a maioria de suas tecnologias Java sob a Licença Pública Geral GNU. Enquanto isso, outros desenvolveram implementações alternativas dessas tecnologias da *Sun*, como o *GNU Compiler for Java* (compilador de *bytecode*), *GNU Classpath* (bibliotecas padrão) e *IcedTea -Web* (plug-in de navegador para *applets*) (GOSLING, 2015).

As versões mais recentes são Java 13, lançado em setembro de 2019, e Java 11, uma versão de suporte a longo prazo (LTS) atualmente suportada, lançada em 25 de setembro de 2018; A Oracle lançou para o Java 8 LTS legado a última atualização pública gratuita em janeiro de 2019 para uso comercial, enquanto ainda suportará o Java 8 com atualizações públicas para uso pessoal até pelo menos dezembro de 2020. A *Oracle* (e outros) recomendam altamente a desinstalação antiga versões do Java devido a sérios riscos devido a problemas de segurança não resolvidos. Como o Java 9 (e 10 e 12) não é mais suportado, a *Oracle* aconselha seus usuários a migrarem imediatamente para a versão mais recente (atualmente Java 13) ou uma versão LTS (SILVA, 2015).

A linguagem Java é um pilar fundamental do Android, um sistema operacional móvel de código aberto. Embora o Android, construído no kernel do Linux, seja escrito principalmente em C, o Android SDK usa a linguagem Java como base para aplicativos Android, mas não usa nenhum de seus GUI, SE, ME ou outros padrões Java estabelecidos. A linguagem de *bytecode* suportada pelo Android SDK é incompatível com o *bytecode* Java e é executada em sua própria máquina virtual, otimizada para dispositivos com pouca memória, como smartphones e tablets. Dependendo da versão do Android, o *bytecode* é interpretado pela máquina virtual *Dalvikou* compilado em código nativo pelo *Android Runtime* (LUCKOW, 2015).

O Android não fornece a biblioteca padrão completa do Java SE, embora o Android SDK inclua uma implementação independente de um grande subconjunto dela. Ele suporta Java 6 e alguns recursos do Java 7, oferecendo uma implementação compatível com a biblioteca padrão (*Apache Harmony*).

O uso da tecnologia relacionada ao Java no Android levou a uma disputa legal entre Oracle e Google. Em 7 de maio de 2012, um júri de São Francisco descobriu que, se as APIs pudessem ter direitos autorais, o Google violaria os direitos autorais da Oracle pelo uso de Java em dispositivos Android. O juiz distrital William Haskell Alsup decidiu em 31 de maio de 2012 que APIs não podem ser protegidas por direitos autorais, mas isso foi revertido pelo Tribunal de Apelações dos Estados Unidos para o Circuito Federal em maio de 2014 (GOSLING, 2015).

Em 26 de maio, 2016, o tribunal distrital decidiu a favor do Google, declarando que a violação de direitos autorais da API Java no Android constitui uso justo. Em março de 2018, essa decisão foi revogada pelo Tribunal de Apelações, que enviou o caso de determinação dos danos ao tribunal federal de São Francisco. O Google entrou com uma petição de ordem de *certiorari* no Supremo Tribunal dos Estados Unidos em janeiro de 2019 para contestar as duas decisões que foram tomadas pelo Tribunal de Apelações em favor da Oracle (BOHN, 2019).

2.4.2 Framework

No desenvolvimento de aplicativos, desenvolvimento de *software* e desenvolvimento *web*, as estruturas são bibliotecas de códigos com funções comuns definidas. Dessa forma, o usuário precisa apenas de uma pequena linha de código para realizar o que de outra forma precisaria de longos trechos de código. Isso ajuda a acelerar o processo de desenvolvimento para programadores (SILVA, 2015).

Uma maneira de visualizar uma estrutura é imaginar casas pré-fabricadas. Os diferentes módulos são pré-construídos. Eles são simplesmente reunidos no site. Você não está 'construindo' uma porta do zero. Você está apenas pegando uma porta pré-montada e encaixando-a. Isso economiza tempo (MULLEN, et al. 2017).

Um bom exemplo de estrutura seria *o.NET Core*, uma estrutura de código aberto da Microsoft. Essa estrutura é ótima para criar aplicativos para os sistemas operacionais comuns - Windows, Mac OS e Linux - desde que você esteja confortável trabalhando com C # (OGLIARI, et al. 2015).

As estruturas de aplicativos tornaram-se populares com o aumento das interfaces gráficas de usuário (GUIs), uma vez que elas tendiam a promover uma estrutura padrão para aplicativos. Os programadores acham muito mais simples criar ferramentas automáticas de criação de GUI ao usar uma estrutura padrão, pois isso define a estrutura de código subjacente do aplicativo com antecedência. Os desenvolvedores geralmente usam técnicas de programação orientada a objetos (OOP) para implementar estruturas de modo que as partes exclusivas de um aplicativo possam simplesmente herdar das classes existentes na estrutura (DESHPANDE, et al. 2017).

2.4.3 Eclipse Modeling Framework (EMF)

O *Eclipse Modeling Framework* (EMF) é uma estrutura de modelagem baseada em Eclipse e um recurso de geração de código para a construção de ferramentas e outros aplicativos baseados em um modelo de dados estruturado (DESHPANDE, et al. 2017).

Em outras palavras, o projeto EMF é uma estrutura de modelagem e um recurso de geração de código para a construção de ferramentas e outros aplicativos com base em um modelo de dados estruturado. A partir de uma especificação de modelo descrita em XMI, o EMF fornece ferramentas e suporte de tempo de execução para produzir um conjunto de classes Java para o modelo, juntamente com um conjunto de classes de adaptadores que permitem a visualização e edição baseada em comando do modelo e um editor básico (GRANDGEORGE, et al. 2016).

O EMF (núcleo) é um padrão comum para modelos de dados, com base em muitas tecnologias e estruturas. Isso inclui soluções de servidor, estruturas de persistência, estruturas de interface do usuário e suporte para transformações. Dê uma olhada no projeto de modelagem para obter uma visão geral das tecnologias EMF (SILVA, et al. 2015).

A partir de uma especificação de modelo descrita em *XML Metadata Interchange* (XMI), o EMF fornece ferramentas e suporte de tempo de execução para produzir um conjunto de classes Java para o modelo, um conjunto de classes de adaptadores que permitem a visualização e edição baseada em comando do modelo e uma base editor. Os modelos podem ser especificados usando documentos Java, UML, XML ou ferramentas de modelagem anotados e importados para o EMF. Mais importante de tudo, o EMF fornece a base para a

interoperabilidade com outras ferramentas e aplicativos baseados no EMF (BAIG, et al. 2015).

O *Ecore* é o (meta) modelo central no coração da EMF. Permite expressar outros modelos, alavancando suas construções. O *Ecore* também é seu próprio metamodelo (ou seja: o *Ecore* é definido em termos de si mesmo) (PATRICK, et al. 2015).

De acordo com *Ed Merks*, líder do projeto EMF, "*Ecore* é a implementação de referência padrão do EMOF da OMG " (*Essential Meta-Object Facility*). Ainda de acordo com *Merks*, o EMOF foi realmente definido pelo OMG como uma versão simplificada do 'C'MOF mais abrangente, aproveitando a experiência da simplificação bem-sucedida da implementação original do *Ecore* (BAIG, et al. 2015).

O uso do *Ecore* como um metamodelo de base permite que um modelador aproveite todo o ecossistema e ferramentas EMF - na medida em que seja razoavelmente fácil mapear modelos no nível de aplicativo de volta ao *Ecore*. Isso não quer dizer que é uma prática recomendada os aplicativos alavancarem diretamente o *Ecore* como seu metamodelo; em vez disso, eles podem considerar definir seus próprios metamodelos com base no *Ecore* (SILVA, et al. 2015).

3 PROCEDIMENTOS METODOLÓGICOS

3.1 CARACTERIZAÇÃO DO ESTUDO

Esta pesquisa foi desenvolvida através do estudo de caso descritivo, partindo da compreensão geral do assunto buscando descrever os fatos ou situações e proporcionar conhecimento acerca do fenômeno estudado e comprovar ou contrastar relações evidenciadas no caso. Este tipo de estudo tem como finalidade explicar o conteúdo das premissas. De acordo com Gil (2008):

“O estudo de caso é caracterizado pelo estudo profundo e exaustivo de um ou de poucos objetos, de maneira a permitir o seu conhecimento amplo e detalhado, tarefa praticamente impossível mediante os outros tipos de delineamentos considerados.”

Pesquisa descritiva é utilizada para descrever características de uma população ou fenômeno em estudo. Não responde a perguntas sobre como / quando / porque as características ocorreram. Em vez disso, aborda a questão "o quê" (quais são as características da população ou situação que está sendo estudada?) As características usadas para descrever a situação ou população são geralmente algum tipo de esquema categórico também conhecido como categorias descritivas.

Assim, a pesquisa descritiva não pode descrever o que causou uma situação. Assim, a pesquisa descritiva não pode ser usada como base de uma relação causal, em que uma variável afeta outra. Em outras palavras, a pesquisa descritiva pode ter baixa exigência de validade interna.

A pesquisa idealizada, quanto aos objetivos, é de natureza exploratória, com utilização de material literário, a fim de maior familiarização com o assunto estudado. De acordo com Gil (2008, p. 41) “estas pesquisas têm como objetivo proporcionar maior familiaridade com o problema, com vistas a torná-lo mais explícito ou construir hipóteses. Pode-se dizer que estas pesquisas têm como objetivo principal o aprimoramento de idéias ou a descoberta de intuições”. Por isso, pretendeu-se coletar informações disponíveis em livros diversos e materiais adquiridos através de meio eletrônico. Este projeto de pesquisa será realizado pela tipologia não experimental, pois não haverá influência ou manipulação de dados pelo pesquisador, onde as variáveis de interesse do estudo serão observadas ou mensuradas.

De acordo com Coelho (2018). A pesquisa não experimental consiste no estudo de fenômenos sem a intervenção sistemática do pesquisador. Quanto ao tipo de pesquisa será do tipo descritiva pois o objetivo é descrever as características do objeto em estudo, apenas descrevendo a realidade já existente.

Seguindo com o presente trabalho, realizou-se um estudo de caso, o qual fora desenvolvido sob uma abordagem qualitativa, que segundo Michel (2015) é definida, como aquela que se fundamenta na discussão de dados entre duas ou mais pessoas, analisando a partir do significado que estas dão aos seus atos. A autora afirma que tal pesquisa admite a verdade através de análise detalhada. Deve-se considerar que todos, mesmo aqueles menos dotados de conhecimento devem compreender e interpretar o exposto na discussão.

Seus fins, refere-se a uma pesquisa de caráter descritivo, que segundo Gil (2014), visa descrever as características de determinada população, fenômeno ou o estabelecimento de relações entre variáveis. Este tipo de pesquisa tem por finalidade, colocar o pesquisador em um contato direto com tudo o que fora escrito dito ou filmado sobre determinado assunto (MARCONI e LAKATOS, 2011). Segundo Collis e Hussey (2005), a unidade de análise,

pode ser compreendida como o objeto social, a partir do qual o estudo será embasado, ou seja, onde o fenômeno em questão estará sendo observado.

Sendo assim, a presente pesquisa poderá ser classificada como pesquisa de campo analítica, pois visará analisar os dados coletados por meio de um questionário o qual será enviado via e-mail à moradores da cidade de Curitiba. Segundo Michel (2015) a pesquisa de campo consiste na verificação de como a teoria é aplicada na prática e a pesquisa descritiva é definida por Vergara (2009) como aquela que relata as características de determinado fenômeno ou população, podendo relacionar com variáveis e definir sua natureza.

3.2 CAMPO DE ESTUDO

A unidade de análise da pesquisa, se limitará à 100 moradores da cidade Curitiba, tendo como natureza entender a visão deles sobre como a proposta de aplicativo aqui apresentada facilitaria seus treinos.

3.3 INSTRUMENTOS PARA COLETA DE DADOS

Para coleta de dados optou-se pela entrevista através de um questionário semi estruturado, que segundo Michel (2015), é uma conversação profissional entre duas ou mais pessoas, a fim da obtenção de informações a respeito de determinado assunto. A autora ressalta que a entrevista terá um resultado mais satisfatório se os entrevistados pertencerem ao grupo específico do assunto em questão, tendo em vista que esse terá melhor condição de fornecer informações relevantes para o resultado da pesquisa.

4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Como citado na seção acima, o presente estudo de caso contou com 100 entrevistados. A primeira pergunta pela qual eles foram questionados foi: quantas vezes você malha na semana?

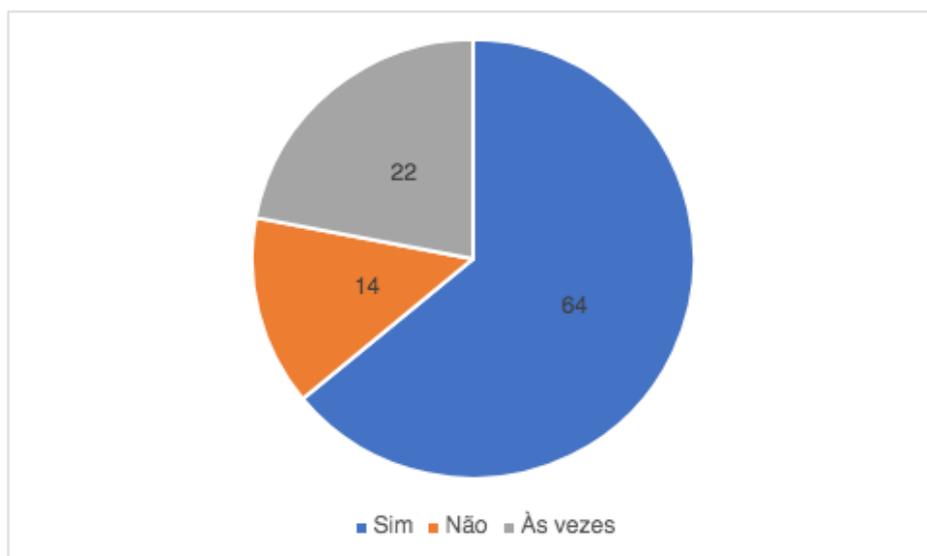
Gráfico 1 - Frequência semanal na academia



Fonte: o autor (2020).

Seguindo, eles foram questionados com a seguinte pergunta: você faz o uso do aparelho celular durante suas atividades dentro do ambiente da academia?

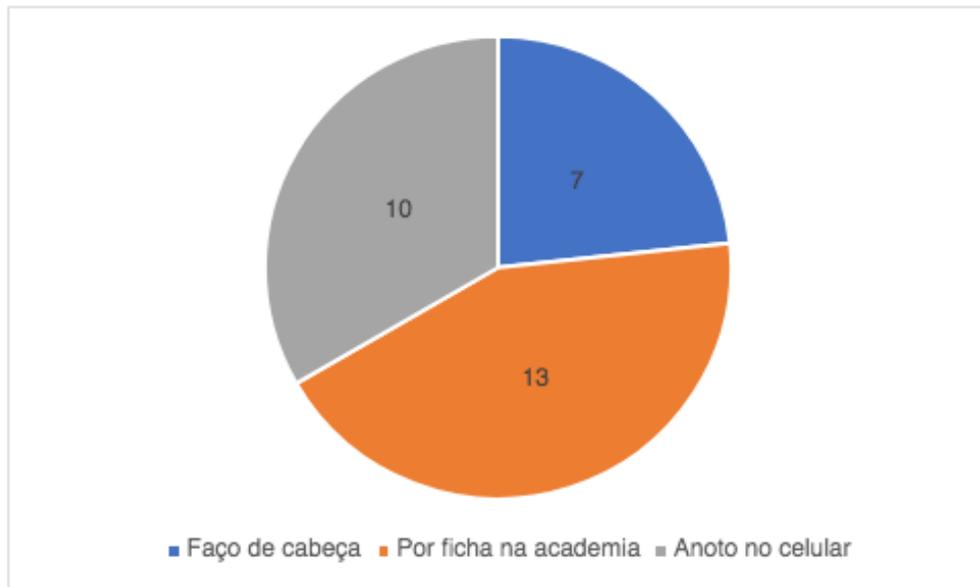
Gráfico 2 - Uso do aparelho celular durante o treino



Fonte: o autor (2020).

A terceira pergunta buscou mensurar como eles controlam seus treinos, como apresenta o gráfico abaixo.

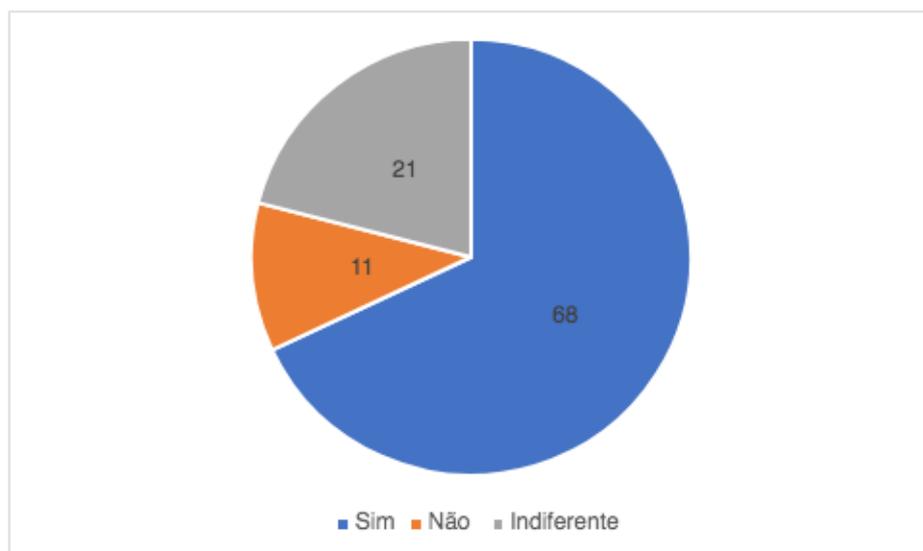
Gráfico 3 - Controle de treinos



Fonte: o autor (2020).

A quarta pergunta pela qual eles foram questionados, foi: você acredita que um aplicativo para cadastro dos treinos seria algo funcional em seu dia-a-dia?

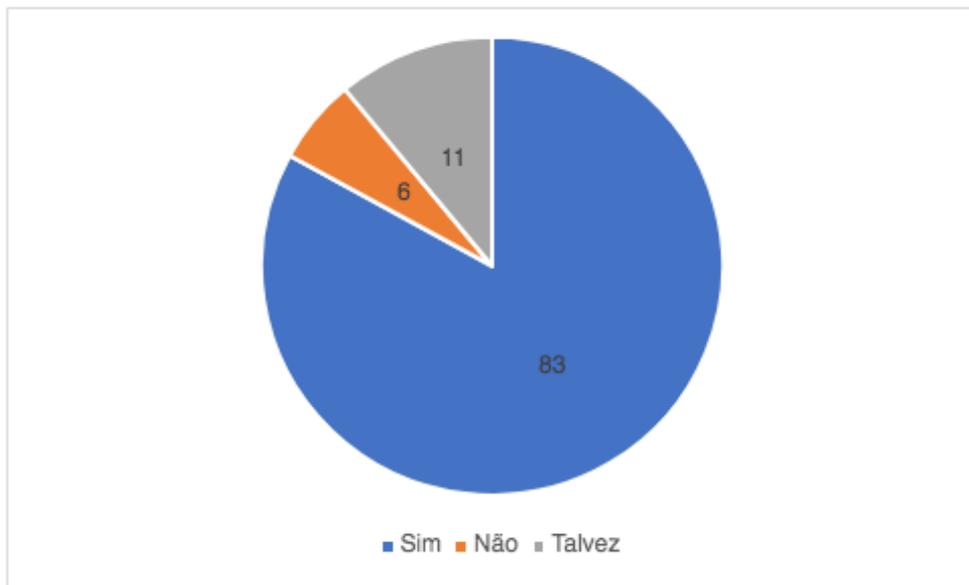
Gráfico 4 - Aplicativo de treinos



Fonte: o autor (2020).

Por fim, a quinta e última pergunta foi: você usaria um aplicativo de cadastro de treinos?

Gráfico 5 - Uso do aplicativo de treinos

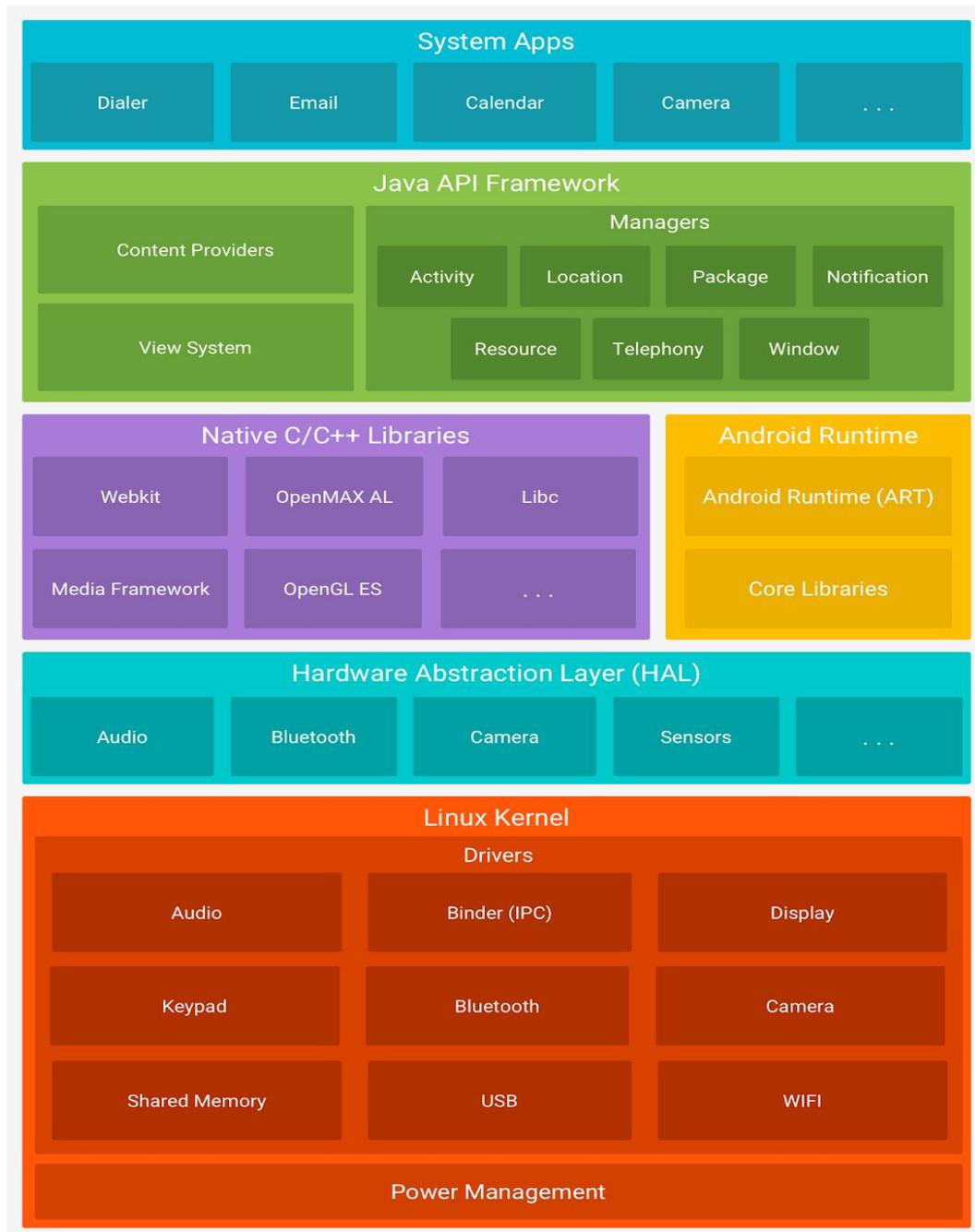


Fonte: o autor (2020).

Tomando por base estas avaliações, pode-se entender que a proposta do desenvolvimento de um aplicativo para uma academia pode vir a ser um diferencial comercial. Dentro desta ideia apresenta-se aqui a referida proposta.

A proposta do aplicativo é baseada em uma plataforma de arquitetura android. Como fora visto, o android é uma pilha de *software* baseado no Linux de código aberto, criada para diversos fatores e com aplicação em diversos dispositivos. Adiante, apresenta-se o um diagrama com a maioria dos componentes utilizados pela plataforma Android.

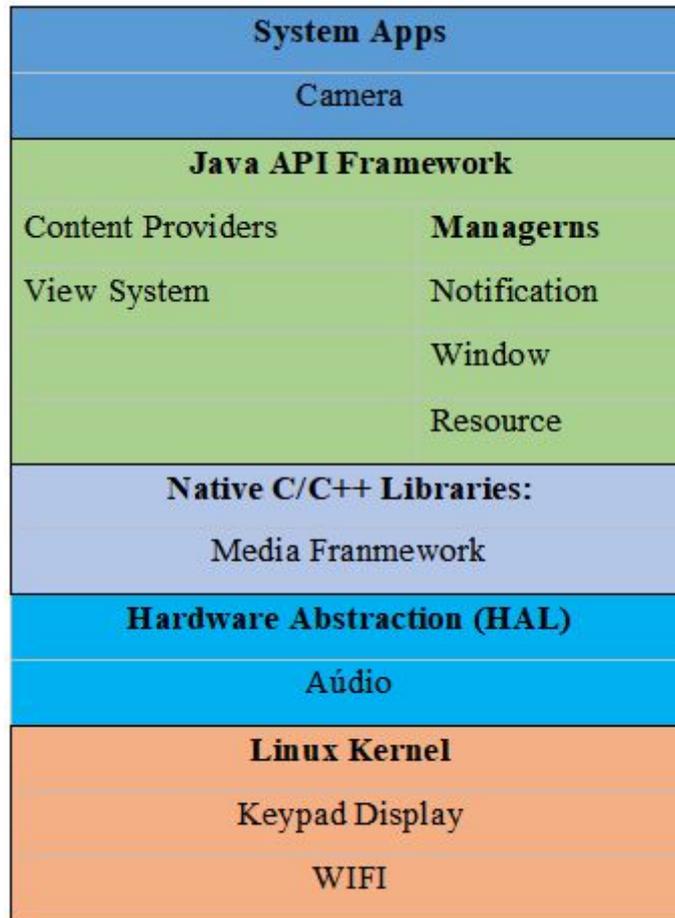
Figura 1 - Principais componentes utilizados pela plataforma Android



Fonte: Botelho (2019).

O objetivo do aplicativo aqui proposto é de apresentar um aplicativo de colaboração interativo, onde os professores e instrutores de uma academia terão a opção de cadastrar as fichas dos treinos de seus alunos, os quais no dia-a-dia conseguirão acessar suas fichas de treinos diários, além de permitir que estes solicitem correções nos treinos, apague atividades indesejadas e acompanhe a evolução de suas atividades. De uma forma simples, a arquitetura do presente projeto ficará da seguinte maneira:

Figura 2 - Arquitetura proposta para o aplicativo



Fonte: o autor (2020).

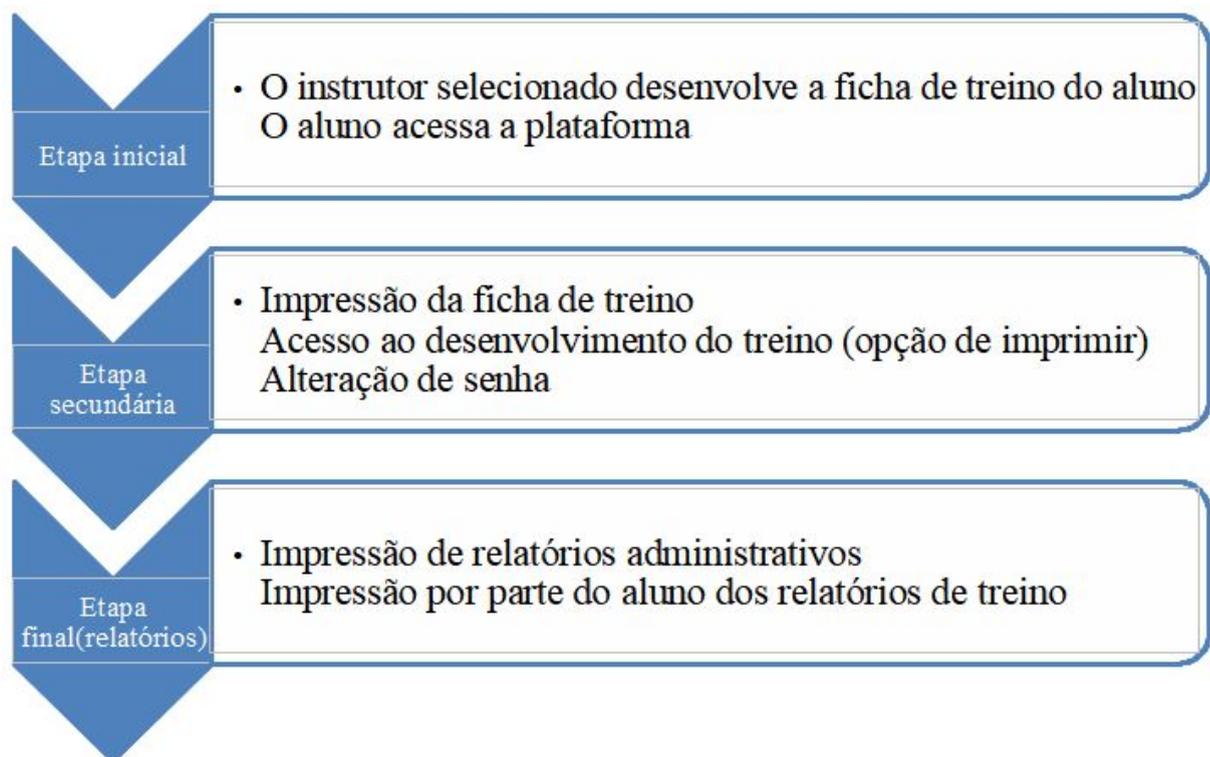
Desta maneira, o aplicativo proposto, resume-se basicamente em um sistema de gerenciamento de tarefas voltado para as tarefas de qualquer academia. Essencialmente, o software de gerenciamento de ginástica e fitness é uma ferramenta de gerenciamento total que ajuda a agilizar as tarefas administrativas a fim de aumentar o envolvimento, economizar tempo, aumentar a receita, melhor retenção de membros e muito mais. O software de negócios de fitness facilita o gerenciamento eficaz da academia e dos membros da equipe. Segundo Laudon e Traver (2017), os sistemas de relatórios gerenciais ajudam na dados necessários aos gerentes para administrar um negócio eficaz.

Os dados podem variar de dados financeiros, número de funcionários, cliente, contas, produtos, ativos do cliente em custódia, desempenho do investimento, etc. O escopo de um sistema de relatórios gerenciais é amplo.

A vantagem do registro no aplicativo proposto é que elimina a necessidade de planilhas de horas que precisam ser preenchidas. Ao usar o registro no aplicativo aqui sugerido, o instrutor e o aluno não precisam se preocupar em preencher formulários impressos sem erros. Após o registro no aplicativo, suas informações serão armazenadas com segurança em sua própria conta digital no servidor da empresa.

Ao salvar essas informações no aplicativo é mais seguro do que guardar um pedaço de papel. Cada conta tem sua própria senha para fins de segurança e o aplicativo é projetado para eliminar o problema de perda de dados. Isso levará a uma prática comercial mais segura e eficiente. Abaixo apresenta-se o diagrama do escopo da solução proposta após o cliente se matricular na academia:

Figura 3 - Estrutura do escopo da solução



Fonte: o autor (2020).

Abaixo apresenta-se a lista de requisitos do sistema

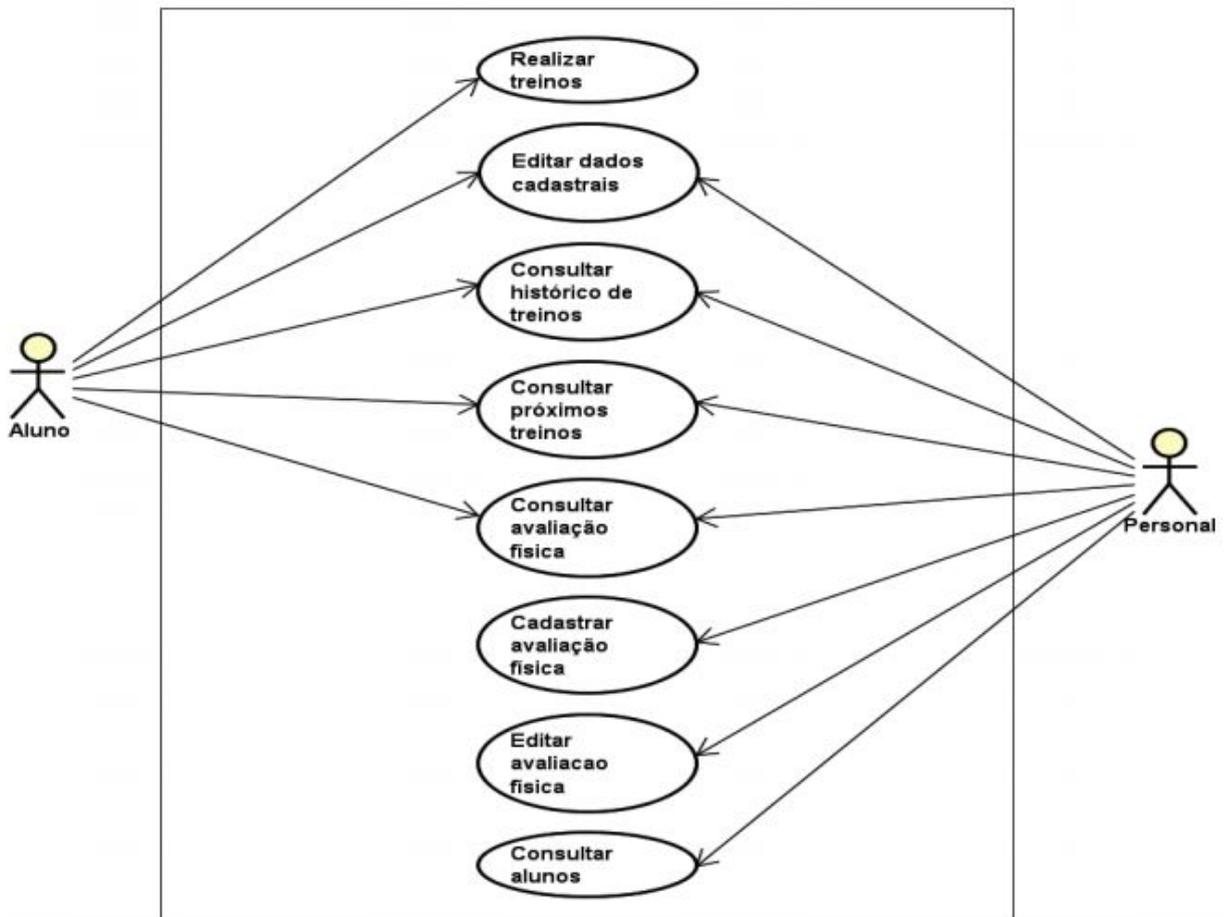
Figura 4 - Diagrama de requisitos



Fonte: o autor (2020).

Seguindo, tem-se o diagrama de casos de uso:

Figura 5 - Diagrama de caso de uso



Fonte: o autor (2020).

Por fim, tem-se as especificações textuais dos casos de uso:

- Realizar treinos: Inicialmente o aluno só estará apto a realizar os treinos se os requisitos executáveis pelo Personal (instrutor) “consultar alunos;” “editar avaliação física;” e “cadastrar avaliação física” estiver validado por este personagem.
- Editar dados cadastrais: Esta função é habilitada para ambos os personagens, podendo o Personal e o aluno editá-la. Salienta-se que na tela do aluno, ele não tem poder para editar a ficha de treino.



- Consultar histórico de treinos: Função habilitada para ambos os personagens. O aluno pode consultar seu histórico e evolução nos treinos, e o Personal pode consultar o desempenho e dedicação do aluno no treino.
- Consultar próximos treinos: Função habilitada para ambos os personagens. Ferramenta simples de gerenciamento de treino.
- Consultar avaliação física: Aluno e instrutor (Personal) podem consultar a avaliação física que fica registrado na ficha do aluno.
- Cadastrar avaliação física: Função habilitada somente para o Personal. Esta serve para atualização ou cadastro de novo usuário (aluno). Tal ferramenta aparecerá para o aluno na ferramenta anterior a esta.
- Editar avaliação física: Função habilitada somente para o Personal. Tal ferramenta é uma tela simples, podendo ser editada apenas pelo Personal.
- Consultar alunos: Função habilitada somente para o Personal. Ferramenta totalmente administrativa com a finalidade

5 CONSIDERAÇÕES FINAIS

O presente trabalho apresentou uma proposta de um aplicativo para academias para dispositivos android. O método adotado na formulação deste trabalho, encontra-se em concordância com a proposta de estudo, a qual encontra-se adequada por meio dos objetivos a serem alcançados. O desenvolvimento da ciência tem como base o alcance de resultados que permite validar hipóteses sobre determinado acontecimento ou fato, presente em nossas vidas, ou não.

Por meio desta pesquisa pode-se entender e compreender a importância que o desenvolvimento de um aplicativo tem, para tal procedimento é necessário conhecer e compreender como se dá todo o processo. Assim, por meio desta pesquisa, pode-se concluir que os dispositivos móveis juntamente com a linguagem android estão em constante evolução, o que com o passar dos anos vem impactando diretamente toda a sociedade e seus meios convencionais de trabalho.

Conclui-se então que tal proposta se apresenta válida e eficaz para em um futuro desenvolver tal produto. Salienta-se que antes de tal desenvolvimento se deve realizar um estudo de campo, a fim de validar as ideias e objetivos aqui apresentados. Em suma, entendeu-se que a efetividade de tal produto irá suprir as necessidades pela qual o problema de pesquisa fora proposto.

Por fim, o presente trabalho deixa o tema em aberto, propondo que no futuro se realize uma nova pesquisa, com a finalidade de contextualizar os temas aqui abordados. Juntamente com esta nova pesquisa, sugere-se a realização de um estudo de caso, para o qual propõe-se uma análise de mercado quanto a aceitação de tal aplicativo. Juntamente com esta análise de mercado sugere-se que se valide a proposta aqui apresentada e aplique uma análise de campo prática, com a finalidade de verificar sua efetivação.

REFERÊNCIAS

AMADEO, R. **O "Project Treble" do Google resolve um dos muitos obstáculos de atualização do Android** . REVISTA Ars Technica Condé Nast. (2017).

ANDROID. **Android Interfaces and Architecture** | Android Open Source Project. 2019. Disponível em: <<http://source.android.com/devices/index.html>>. Acesso em 12 de fevereiro de 2019.

BAIG, M. M.; GHOLAMHOSSEINI, H.; CONNOLLY, M. J. **Mobile healthcare applications: system design review, critical issues and challenges**. Australasian Physical & Engineering Sciences in Medicine, v. 38, n. 1, p. 23–38, 2015.

BEZERRA, P. T.; SCHIMIGUEL, J. **Desenvolvimento de aplicações mobile cross platform utilizando phonegap**. Eumed. 2016.

BJORN-HANSEN, A.; GROENLI, T.; GHINEA, G. **A survey and taxonomy of core concepts and research challenges in cross-platform mobile development**. 2018.

BOHN, Dieter (2019). **"Android 10 lançado oficialmente para telefones Google Pixel"** . The Verge. (2019).

CLARK, Jack; MORITZ, Scott (2016). **"O Google aumenta a pressão sobre os parceiros Tardy na atualização do Android"** . Bloomberg LP (2016).

DEHLING, T. **Exploring the Far Side of Mobile Health: Information Security and Privacy of Mobile Health Apps on iOS and Android**. JMIR mHealth and uHealth, v. 3, n. 1, p. e8, 2015.

DESHPANDE, S.; CHAHANDE, J.; RATHI, A. **Mobile learning app: A novel method to teach clinical decision making in prosthodontics**. Education for Health: Change in Learning and Practice, v. 30, n. 1, p. 31–34, 2017.

EADICICCO, Lisa (2015). **"A ascensão do ANDROID: como uma startup decadente se tornou a maior plataforma de computação do mundo"** . Business Insider . Axel Springer SE.

EL-KASSAS, W. S. **Taxonomy of cross-platform mobile applications development approaches**. Ain Shams Engineering Journal, Elsevier BV, v. 8, n. 2, p. 163–190, jun 2017.

- GARGENTA, M.; NAKAMURA, M. **Learning Android**. Sebastopol, CA: O'Reilly Media, 2015.
- GOSLING, James; ALEGRIA, Bill; STEELE, Guy; BRACHA, Gilad; BUCKLEY, Alex (2015). **A especificação da linguagem Java Java SE 8** ed. 2015.
- GRANDGEORGE, M.; MASATAKA, N. **Atypical Color Preference in Children with Autism Spectrum Disorder**. v. 7, n. December, p. 1–5, 2016.
- HARRISON, G. **Next Generation Databases**. [S.l.]: Apress, 2015.
- LECHETA, R. R. **Google Android - Aprenda a criar aplicações para dispositivos móveis com o Android SDK**. 5. ed. [S.l.]: Novatec Editora, 2015.
- LUCKOW, Décio H.; MELO, Alexandre A. **Programação JAVA Para Web**. São Paulo: Novatec, 2015.
- LUPTON, D.; JUTEL, A. **Social Science & Medicine** “ It is like having a physician in your pocket ! ” A critical analysis of self diagnosis smartphone apps. *Social Science & Medicine*, v. 133, n. January 2014, p. 128–135, 2015.
- MELLO, J. A. C.; SILVA, S. A. P. **Competências do gestor de academias esportivas**. **Motriz**: Revista de Educação Física, Rio Claro, v.19, n.1, p.74- 83, Mar. 2013.
- MULLEN, S.; PAGOTO, S. **Evaluating and selecting mobile health apps**: strategies for healthcare providers and healthcare organizations. n. May 2012, 2017.
- OGLIARI, Ricardo da Silva; BRITO, Robison Cris. **Android - do básico ao avançado**. Rio de Janeiro: Editora Ciência Moderna Ltda., 2015.
- PATRICK, J. R. **How Health will spur consumer-led healthcare**. *mHealth*, v. 1, p. 14, 2015.
- SARTOR, LUIZ , BRISOLARA; ANDERSON; CORREA, ULISSES ; BECK, ANTONIO CARLOS SCHNEIDER. **AndroProf: A Profiling Tool for the Android Platform**. *Computing Systems Engineering (SBESC)*, 2015 III Brazilian Symposium on, pp. 23-28. IEEE, 2015.



SILVA, M. M. DA; SANTOS, M. T. P. **Os Paradigmas de Desenvolvimento de Aplicativos para Aparelhos Celulares**. T.I.S - Tecnologias, Infraestrutura e Software - UFSCar, v. 3, n. 2, p. 162– 170, 2015.

SILVA, R.P. **Suporte ao desenvolvimento e uso de frameworks e componentes**. Tese de Doutorado, UFRS, 2015.

TANENBAUM, A. S. **Sistemas Operacionais Modernos**. [S.l.]: Pearson, 2015.

ANEXOS



1) Quantas vezes você malha na semana?

Um dia
Dois dias
Três ou mais dias
Todos os dias da semana

2) Você faz o uso do aparelho celular durante suas atividades dentro do ambiente da academia?

Sim
Não
Às vezes

3) Como você controla seu treino?

Faço de cabeça
Por ficha na academia
Anoto no celular

4) Você acredita que um aplicativo para cadastro dos treinos seria algo funcional em seu dia-a-dia?

Sim
Não
Indiferente

5) Você usaria um aplicativo de cadastro de treinos?

Sim
Não
Talvez